INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2003; **1**:1–21 *Prepared using* nmeauth.cls *[Version: 2002/09/18 v2.02]*

Estimating the critical time-step in explicit dynamics using the Lanczos method

J. R. Koteras[1] and R. B. Lehoucq[2]

[1] *Computational Solid Mechanics & Structural Dynamics, Sandia National Laboratories, MS 0380, P.O.Box 5800, Albuquerque, NM 87185-0380 (*`jrkoter@sandia.gov`*)*

[2] *Computational Mathematics & Algorithms, Sandia National Laboratories, MS 1110, P.O.Box 5800, Albuquerque, NM 87185-1110 (*`rblehou@sandia.gov`*).*

SUMMARY

The goal of our paper is to demonstrate the cost-effective use of the Lanczos method for estimating the critical time step in an explicit, transient dynamics code. The Lanczos method can give a significantly larger estimate for the critical time-step than an element-based method (the typical scheme). However, the Lanczos method represents a more expensive method for calculating a critical time-step than element-based methods. Our paper shows how the additional cost of the Lanczos method can be amortized over a number of time steps and lead to an overall decrease in run-time for an explicit, transient dynamics code. We present an adaptive hybrid scheme that synthesizes the Lanczos-based and element-based estimates and allows us to run near the critical time-step estimate provided by the Lanczos method.

KEY WORDS: Eigenvalues, Lanczos iteration, critical time step, explicit dynamics

1. Introduction

Codes using explicit time integration techniques are important for simulating transient dynamics problems involving large deformation of solids with various nonlinear effects (contact, nonlinear materials, element death, etc.). The second order central difference operator used in explicit codes is stable if the time step is no larger than the critical time step. For most problems in solid mechanics, the critical time step is extremely small and the number of time steps required for a typical analysis is quite large. Therefore, the accurate, efficient, and reliable calculation of the critical time step is of fundamental importance.

The element-based method [1] is an efficient method for producing a critical time step estimate at every time step. However, it can produce a conservative estimate for the critical

*Correspondence to: R. B. Lehoucq, Sandia National Laboratories

Contract/grant sponsor: Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy; contract/grant number: DE-AC04-94AL85000

Received April 6, 2006
Revised April 6, 2006

time step in many cases. The Lanczos [5] method is a reliable procedure for producing a time step that is the theoretical maximum value for a structure and is usually much better than the element-based estimate. The cost of obtaining a Lanczos based estimate will not offset the cost benefit of the increased value for the critical time step. Therefore, it is not feasible to call the Lanczos method at every explicit dynamics time step. In this paper we outline a cost-effective method for utilizing the Lanczos method (together with an element-based scheme) for the critical time step estimation. Therefore,

There are two crucial issues that must be addressed in combining the Lanczos-based and element-based estimates. First, the Lanczos-based time step estimate must be used for two to three times the number of explicit time integration steps required to recover the cost of the Lanczos method if we are to see a noticeable reduction in overall computation times for a problem. Second, the Lanczos method provides an overestimate of the critical time step and so we need an effective scheme to scale back the Lanczos-based critical time step estimate.

We also remark that in addition to the increased efficiency that can be achieved with the Lanczos-based time step, we also have the added benefit of increased accuracy. For explicit transient dynamic codes, using a time step as close as possible to the critical time step [4] gives the most accurate answer. Reducing the time step in an explicit transient dynamics code actually increases the error.

Our paper is organized as follows. Section 2 discusses the critical time step and motivates a Lanczos-based estimate. The Lanczos iteration and method are briefly introduced in section 3. A cost benefit analysis of the element-based and Lanczos-based approximations to the critical time is considered in section 4. A practical implementation within an explicit dynamics code is the subject of section 5. Several numerical examples are presented in section 6, and we provide our conclusions in section 7.

2. Critical time step

Let $\mathbf{K}$ and $\mathbf{M}$ be the stiffness and mass matrices arising in an explicit dynamics simulation so that $\mathbf{M}$ is a diagonal matrix due to mass lumping. The critical time step for second order central time differencing is bounded from above by $2\omega_{\max}^{-2}$ where $\omega_{\max}^2$ is the largest eigenvalue of the generalized eigenvalue problem

$$\mathbf{K}\mathbf{u} = \mathbf{M}\mathbf{u}\omega_{\max}^2, \quad \left(\mathbf{K}, \mathbf{M} \in \mathbb{R}^{n\times n}\right), \tag{1}$$

where we assume that $\omega_{\max}^2$ is positive. An inexpensive [2] upper bound to ω^2 is given by the maximum element eigenvalue $\omega_{\max,e}^2$ over all the element eigenvalue problems

$$\mathbf{K}^e\mathbf{u}^e = \mathbf{M}^e\mathbf{u}^e\omega_e^2, \quad \left(\mathbf{K}^e, \mathbf{M}^e \in \mathbb{R}^{n^e\times n^e}\right), \tag{2}$$

where $n^e \ll n$. Therefore, $\omega_{\max,e}^{-2} \leq \omega_{\max}^{-2}$ and we have a lower bound for the critical time step. The maximal element eigenvalue is typically computed analytically [1] for the finite elements that are typically used in transient dynamics.

The Lanczos method rapidly provides a lower bound $\omega_{\max,L}^2$ to $\omega_{\max}^2$ so that

$$\omega_{\max,e}^{-2} \leq \omega_{\max}^{-2} \leq \omega_{\max,L}^{-2}. \tag{3}$$

In fact, the Lanczos iteration is sharp so that $\omega_{\max}^{-2} \lessapprox \omega_{\max,L}^{-2}$ so that with care, an excellent approximation to the critical time step is computed for a modest cost. This approximation may be dramatically superior to the standard element based estimate. The details of a careful use of the Lanczos-based estimate is the subject of section 5.

3. Lanczos iteration

The Lanczos reduction rapidly provides approximations to the extremal eigenvalues of a symmetric $\mathbf{A} \in \mathbb{R}^{n\times n}$, in particular the largest in magnitude eigenvalue. Suppose that

$$\mathbf{A}\mathbf{Q}_j = \mathbf{Q}_j\mathbf{T}_j + \mathbf{f}_j\mathbf{e}_j^T, \tag{4}$$

is a Lanczos reduction of length j where $\mathbf{f}_j \in \mathbb{R}^n$, and $\mathbf{e}_j \in \mathbb{R}^j$ contains column j of the identity matrix $\mathbf{I}_n \in \mathbb{R}^{n\times n}$. If we denote

$$\mathbf{T}_j = \begin{pmatrix} \alpha_1 & \beta_2 & \cdots & 0 \\ \beta_2 & \alpha_2 & \cdots & 0 \\ \vdots & & \ddots & \beta_j \\ 0 & \cdots & \beta_j & \alpha_j \end{pmatrix}, \quad \alpha_i, \beta_i \in \mathbb{R}$$

and

$$\mathbf{Q}_j = \begin{pmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_j \end{pmatrix}, \quad \mathbf{q}_i \in \mathbb{R}^n$$

then the familiar Lanczos three-term recurrence is recovered by equating column j of (4) to obtain

$$\mathbf{f}_j = \mathbf{A}\mathbf{q}_j - \mathbf{q}_j\alpha_j - \mathbf{q}_{j-1}\beta_{j-1}^T. \tag{5}$$

Furthermore, because of the orthonormality of $\mathbf{Q}_j$, we have

$$\alpha_j = \mathbf{q}_j^T\mathbf{A}\mathbf{q}_j, \tag{6a}$$

$$\mathbf{q}_{j+1}\beta_{j+1} = \mathbf{f}_j, \tag{6b}$$

$$\mathbf{q}_i^T\mathbf{f}_j = 0, \quad i = 1, \ldots, j \tag{6c}$$

and so $\mathbf{q}_{j+1} = \mathbf{f}_j\beta_{j+1}^{-1}$, where we assume that β_{j+1} is non-zero. We define a *Lanczos iteration* to be that computing $\mathbf{A}\mathbf{q}_j, \alpha_j, \beta_{j+1}$, and $\mathbf{f}_j$. We define the *Lanczos method* that of computing m iterations and computing the largest in magnitude eigenvalue of $\mathbf{T}_m$.

The largest eigenvalue of the symmetric tridiagonal matrix $\mathbf{T}_j$ approximates the largest in magnitude eigenvalue of $\mathbf{A}$. We can determine the quality of the approximation produced by an eigenpair of $\mathbf{T}_j$. If we postmultiply (4) by $\mathbf{s}$ where $\mathbf{T}_j\mathbf{s} = \mathbf{s}\theta$ (and $\|\mathbf{s}\| = 1$), then

$$\mathbf{A}(\mathbf{Q}_j\mathbf{s}) - (\mathbf{Q}_j\mathbf{s})\theta = \mathbf{f}_j(\mathbf{e}_j^T\mathbf{s}). \tag{7}$$

In words, the residual of the approximate eigenpair $(\mathbf{Q}_j\mathbf{s}, \theta)$ is proportional to $\mathbf{f}_j$ (note that $\mathbf{e}_j^T\mathbf{s}$ is notation for the last component of $\mathbf{s}$). The implication is that we can easily monitor the quality of the approximation produced by the Lanczos method. If θ is the largest in magnitude eigenvalue of $\mathbf{T}_j$, then $\theta \le \omega_{\max}^2 \le \|\mathbf{f}_j\|_2\,|\mathbf{e}_j^T\mathbf{s}| + \theta$ (see [7] for a discussion). Hence,

$$\frac{1}{\|\mathbf{f}_j\|_2\,|\mathbf{e}_j^T\mathbf{s}| + \theta} \le \omega_{\max}^{-2} \le \frac{1}{\theta}. \tag{8}$$

We also remark that the norm of the residual is a non-increasing function of j; again see [7].

The Lanczos iteration is adapted for computing the largest eigenvalues of (1) by replacing $\mathbf{A}$ with $\mathbf{M}^{-1}\mathbf{K}$ and computing an $\mathbf{M}$-orthonormal $\mathbf{Q}_j$. This orthonormality is needed so that $\mathbf{M}^{-1}\mathbf{K}$ is symmetric in the inner product induced by $\mathbf{M}$. See [6, 7] for further discussion and implementations.

The cost of a careful implementation of a Lanczos iteration, $j > 1$, is one matrix-vector product with $\mathbf{K}$ and $\mathbf{M}^{-1}$, and two vector products and vector subtractions. Within an explicit dynamics code, the cost of computing a Lanczos vector is approximately the cost of an internal force calculation, represented by the matrix-vector product $\mathbf{K}\mathbf{q}_j$. Therefore, we approximate the cost of computing the Lanczos-based time step estimate as

$$m\tau \tag{9}$$

where m denotes the number of Lanczos iterations and τ represents the CPU (central processor unit) time needed for an element-based explicit dynamics time integration step.

The Lanczos method only requires knowledge of $\mathbf{K}$ via its application on a vector. If internal force calculations are used for the needed matrix-vector products, the Lanczos vectors $\mathbf{q}_j$ are scaled so that they represent velocities associated with small strain. When these scaled vectors are sent to the internal force calculation, the internal force calculation becomes a matrix-vector product with a (constant) tangent stiffness matrix $\mathbf{K}_T$.

4. Cost benefit analysis

This section provides a simple model for assessing the cost of using the Lanczos method for computing an estimate of the critical time step. We assume that Lanczos-based time step is valid for n_L time integration steps. We address the important issue of the adapting the time step when we present the details for practical use of the Lanczos method in a subsequent section.

Denote by Δt_L and Δt_e the time steps estimate of the critical time step computed by the Lanczos and element-based methods, where the ratio ρ of Δt_L to Δt_e is at least as large as one because of (3). After n_L time steps, the dynamics simulation is advanced in time $n_L \Delta t_L$. Let n_e be the number of element-based time steps so that $n_e \Delta t_e \le n_L \Delta t_L < (n_e + 1)\Delta t_e$. In terms of ρ, we have the relationship

$$n_e \le \rho n_L < n_e + 1, \tag{10}$$

so bounding the number of Lanczos-based explicit integration steps in terms of ρ and the number of element-based integration steps.

Let us examine the computational costs in terms of CPU time in performing the above n_L and n_e integration steps. Denote by τ the CPU time for an element-based time integration step and assume that it is dominated by the cost of an internal force calculation. Using (10), the CPU time of n_L time integration steps is

$$(n_L + m)\tau, \tag{11}$$

and the CPU time of n_e time integration steps is $n_e\tau$. Equating these two CPU times, determines when the cost of both approaches is equivalent and results in the relationship

$$\hat{n}_e = m + \hat{n}_L. \tag{12}$$

Using (12) within (10) gives

$$\frac{m}{\rho - 1} \le \hat{n}_L < \frac{m+1}{\rho - 1} \tag{13}$$

so bounding the minimum number of Lanczos-based time integration steps in terms of the number of Lanczos iterations and ρ so that the cost of the computing the Lanczos-based time step is amortized.

Our cost benefit analysis provides the "break-even" point at which the Lanczos method becomes cost-effective by overcoming the associated overhead. For example, let $\rho = 1.25$ and $m = 20$ so that $\hat{n}_L$ is bounded from below by 80, and by (11) $\hat{n}_e = 100$. Hence, the time integrations with the Lanczos-based and element-based estimates of the critical time step give the same simulation time for the same CPU time. If we use the Lanczos-based time step Δt_L for more than 80 time integrations steps, then the Lanczos-based approach is cost-effective.

A Lanczos-based critical time estimate is cost effective if m is small and ρ is not close to one. The size of m is dependent upon the ability of the Lanczos method to rapidly provide an accurate approximation to $\omega^2_{\max}$. If ρ approaches one, then the Lanczos-based critical time step approaches the element-based critical time step, implying that $\hat{n}_L$ must increase to offset the cost of the m Lanczos iterations. Section 6 demonstrate that m is small and that ρ is not close to one for realistic problems.

Our section ends by considering the additional cost involved with contact. The addition of contact to an analysis can add computational costs to a time step that are as large as or larger than the internal force calculations. Therefore, for an analysis with contact, running at a larger time step than the element-based estimate can have an even greater impact on reducing CPU time for an analysis.

The above analysis is easily extended to the case where we have contact. If the CPU time of contact over a time step is some multiple γ of τ, then in analogy to (12) and (13), we have

$$(1+\gamma)\hat{n}_e = m + (1+\gamma)\hat{n}_L, \tag{14}$$

and

$$\frac{m}{(\rho - 1)(1+\gamma)} \le \hat{n}_L < \frac{m+1+\gamma}{(\rho - 1)(1+\gamma)} \tag{15}$$

Again, for example, let $\rho = 1.25$ and $m = 20$ and assume the computational cost of contact calculations is the same as an internal force calculation so that $\gamma = 1$. Hence, the break-even point is $\hat{n}_L = 40$ and $\hat{n}_e = 50$. The additional cost of the contact calculations within the time integration reduces the break-even point over that with no contact ($\gamma = 0$).

5. Using the Lanczos-based estimate

The previous section shows how the repeated use of a Lanczos-based time step estimate could be cost-effective within an explicit transient dynamics simulation. This section presents an adaptive scheme that combines the Lanczos-based estimate with an element-based estimate of the critical times-step over a number of explicit time integration steps.

Section (2) explained that the Lanczos method provides an approximation to the maximum eigenvalue of (1) from below so overestimating the critical time step. Therefore, we scale back the Lanczos-based time. The scheme to determine a scaled-back value employs the element-based time step estimate. Again, let Δt_L and Δt_e be the time steps computed by the Lanczos and element-based methods. The scaled back estimate for the critical time step, Δt_s, is computed from the equation

$$\Delta t_s = \Delta t_e + f_s(\Delta t_L - \Delta t_e), \tag{16}$$

where f_s is a scale factor. (The value for f_s ranges from 0.9 to 0.95 for our problems—a rigorous estimate can be made by using (8).) This value of f_s results in Δt_s close to and slightly less than the critical time step. Once Δt_s is determined, the ratio

$$t_r = \frac{\Delta t_s}{\Delta t_e}$$

is computed. This ratio is then used to scale subsequent element-based estimates for the critical time step. If $\Delta t_{e(n)}$ is the n^{th} element-based time step after the time step where the Lanczos method is computed, then the n^{th} time step computed is

$$\Delta t_{(n)} = t_r \Delta t_{e(n)}. \tag{17}$$

The ratio t_r is used until the next call to the Lanczos method. The next call to the Lanczos method is controlled by one of two mechanisms. First, the user can set the frequency with which the Lanczos method is called. The user can set a parameter so that the Lanczos method is called only once every n time steps. This number remains fixed throughout an analysis. Second, the user can control when the Lanczos method is called based on changes in the element-based time step. For this second method, the change in the element-based critical time step estimate is tracked. At the n^{th} step after the call to the Lanczos iteration, the element-based time step is $\Delta t_{e(n)}$. If the value

$$\frac{|\Delta t_{e(n)} - \Delta t_e|}{\Delta t_e} \tag{18}$$

is greater than some limit set by the user, then the Lanczos method is called. If there is a small, monotonic change in the element-based time step over a large number of time integration steps, this second mechanism will result in the Lanczos method being computed. If there is a large, monotonic change in the element-based critical time step over a few time steps, the Lanczos method will also be called.

These two mechanisms for calling the Lanczos method may be combined resulting in an adaptive scheme for estimating the critical time step during an explicit transient dynamics simulation. For example, suppose the second mechanism, the mechanism based on a change in the element-based time step, results in a call to the Lanczos method. This resets the counter for the first mechanism, the mechanism using a set number of time steps between calls to the Lanczos iteration.

6. Numerical experiments

This method for reusing a Lanczos-based time step estimate has been implemented in Presto [3], and employed within a number of explicit dynamics simulations. We discuss several of these examples.

6.1. Example one

The Lanczos method has been used to obtain a critical time step estimate for a cubic block consisting solely of cubic elements—a 10 × 10 × 10 mesh of eight-node hexahedral elements. We know that, for a cubic eight-node hexahedral element, the element-based estimate is conservative by a factor of $1/\sqrt{3}$. The Lancozs method yields a critical time estimate for this mesh that is $\rho = \sqrt{3}$ (approximately 1.732) times larger than the element-based estimate. This is done by using 20 Lanczos vectors.

6.2. Example two

Critical time step estimates were made for two mechanical systems. The systems consisted of cylindrical metal cans containing a variety of components. Some of these components have relatively simple geometries, while other components have complex shapes. A number of the components with complex shapes are a foam material used to absorb impact loads. One component was modeled with approximately 250,000 degrees of freedom, and the other one was modeled with approximately 350,000 degrees of freedom. For both of these models, a good estimate for the maximum eigenvalue was obtained with the Lanczos method by computing only twenty Lanczos vectors. For the model with 250,000 degrees of freedom, an actual analysis was run. The value for ρ for this problem was 1.83. The break-even point for this case ($n_L = 20$ and $\rho = 1.83$) is $n_e = 45$. It was possible to use the same scale factor for 1700 time steps for this analysis, which is well above the break-even point. The extended use of the Lanczos based estimate reduced the computation cost by over 56%.

6.3. Example three

A study of a large-scale model involving 1.7 million nodes (5.1 million degrees of freedom) showed that only 45 Lanczos vectors were required to obtain a good estimate of the maximum eigenvalue. The value of ρ for this problems was 1.2. Use of this Lanczos based estimated for this problem would be extremely cost-effecttive.

7. Conclusions

Our paper presented a cost-effective use of the Lanczos method for estimating the critical time step in an explicit, transient dynamics code. The Lanczos method can give a significantly larger estimate for the critical time-step than an element-based method (the typical scheme). Our adaptive hybrid scheme synthesizes the Lanczos-based and element-based estimates and allows us to run near the critical time-step estimate provided by the Lanczos method.

Not all problems will lend themselves reuse of one Lanczos-based estimate for thousands of time steps. However, if it is possible to use the Lanczos-based estimate for two to three times the number of time steps required for break-even, we begin to see a noticeable reduction in the total CPU time required for a problem.

In addition, to the increased efficiency we can achieve with the Lanczos iteration, we also have the added benefit of increased accuracy. For explicit transient dynamic codes, using a time step as close as possible to the critical time gives the most accurate answer. Reducing the time step in an explicit transient dynamics code actually increases the error.

Acknowledgments

The authors would like to acknowledge discussions with Samuel W. Key, Arne S. Gullerud, and Martin W. Heinstein (Computational Solid Mechanics and Strucutral Dynamics Department, Sandia National Laboratories, Albuquerque) regarding this work.

REFERENCES

1. D. P. Flanagan and T. Belytschko, *Eigenvalues and stable time steps for the uniform strain hexahedron and quadrilateral*, ASME Journal of Applied Mechanics, 51 (1984), pp. 35–40.
2. B. M. Irons, *Applications of a theorem on eigenvalues to finite element problems*, Tech. Report CR/132/70, University of Wales, Department of Civil Engineering, Swansea, U.K., 1970.
3. J. R. Koteras and A. S. Gullerud, *Presto user's guide version 1.05*, Tech. Report SAND2003-1089, Sandia National Laboratories, Albuquerque, NM, April 2003.
4. R. D. Krieg and S. W. Key, *Transient shell response by numerical time integration*, Int. Journal Num. Meth. Eng., 7 (1973), pp. 273–286.
5. C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Research of the National Bureau of Standards, 45 (1950), pp. 255–282. Research Paper 2133.
6. B. Nour-Omid, *The Lanczos algorithm for solution of large generalized eigenproblems*, in The Finite Element Method: Linear static and dynamic finite element analysis,Thomas J.R. Hughes, Dover, 2000, pp. 582–600. Reprint of material that originally appeared in"The Finite Element Method", Prentice-Hall, 1987, by Thomas J.R. Hughes.
7. B. N. Parlett, *The symmetric eigenvalue problem*, no. 20 in Classic in Applied Mathematics, SIAM, 1998. Reprint of "The symmetric eigenvalue problem", Prentice-Hall, 1980.